

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/269049273>

# Modelling and Using Common Ground in Human-agent Collaboration during Spacecraft Operations

Conference Paper · May 2014

DOI: 10.2514/6.2014-1862

---

CITATIONS

4

---

READS

548

3 authors, including:



[Tim Miller](#)

University of Melbourne

134 PUBLICATIONS 3,882 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Agent-Oriented Software Engineering [View project](#)



Agent Design Verification [View project](#)

# Modelling and Using Common Ground in Human-agent Collaboration during Spacecraft Operations

Jens Pfau\*

*CGI Space, 64295 Darmstadt, Germany*

Tim Miller† and Liz Sonenberg‡

*The University of Melbourne, Melbourne, Victoria, 3010, Australia*

Space operations involve the control of complex technical equipment in highly dynamic and unknown environments. This is a challenging task for human operators. To facilitate this task, mission-critical software systems need to be able to truly engage in joint activities with their human operators; i.e. these systems need to be designed for interdependence with the operators. A key aspect of undertaking joint activities in human teams is the representation and maintenance of common ground—the information that participants in a joint activity share and assume to be shared. It has been proposed that maintenance of common ground is also important in the operation of space software systems, but computational representations of the concept have been rather ad-hoc. In recent work, we introduced the first formal account of common ground, which we based on a detailed analysis of philosophical and psychological literature. In this paper, we analyze how computational representations of common ground can support the interdependence of space software systems with human operators; and we explore how this aspect of joint activities might be implemented computationally. Our analysis is relevant to the design of future mission control systems and to the design of human-agent-robot teams in space operations.

## I. Introduction

Software systems are crucial to space mission operations. The key characteristic of these systems is that they enable their users to control complex technical equipment in a challenging and dynamic environment. Researchers and practitioners recognize that these circumstances require software systems designed for *interdependence* with human operators, i.e. systems that have the capability to engage in joint activities with operators, where the outcomes of actions performed by systems and operators influence each other (Bradshaw et al., 2012). The benefit expected from such systems is a more natural and effective interaction between users and systems.

Naturally, these systems require mechanisms for interpreting and engaging in the processes that characterize human joint activities. This includes, amongst others, the capability to interpret and forecast human intentions, to influence the behavior of human partners through targeted modifications of the situational context, and to maintain a shared understanding and awareness of the situation in which the joint activity takes place. Part of this shared understanding that participants hold is their *common ground* (Clark, 1996). Common ground is the information that people share and mutually recognize to be shared. Maintaining common ground is a key requirement for successful joint activities between human operators and software systems (Bradshaw et al., 2011; Clark, 1996).

There are many concrete examples in which software systems used in space operations should be designed for interdependence with human operators. For example, routine tasks and tasks that are computationally

---

\*jens.pfau@cgi.com

†tmiller@unimelb.edu.au

‡l.sonenberg@unimelb.edu.au

tedious such as the monitoring of spacecraft health could be performed more efficiently through a tailored collaboration between operators and software agents (Fitts, 1951). However, empirical evidence demonstrates that, as these software systems increasingly contribute to decision-making, the reduction in cognitive load made possible by the software system is sometimes counter-acted by added confusion of operators as they seek to understand how the system’s decisions came about (Stubbs et al., 2007). Some authors (e.g. Stubbs et al., 2007) believe that maintaining common ground between decision-support and decision-making systems and their human partners will make the relationship more effective. Likewise, those software agents must be able to infer the common ground they have with human actors to react appropriately to input from their partners. Interdependence and therefore common ground is also crucial in the interaction between robotic systems and their operators. In some cases, robots can be considered as extended sensor systems for operators, and therefore they should contribute to the adjustment of their common ground as sensor information and input from the operator is coming in.

Even though the importance of common ground in those areas has been recognized, there is a lack of an accepted formal model of common ground and informal notions vary substantially (Allan, 2013; Lee, 2001). A formal model, however, is a prerequisite for any computational implementation of common ground. We draw on our earlier work of modal logics of common ground (Pfau et al., 2014). These logics are inspired by informal philosophical and social-psychological work. In particular, one model of common ground accounts for the fact that the common ground of a group of actors in a joint interaction is composed not only of the immediate common ground they have created during the course of their interaction, but also of the common ground that they share due to previous common experiences as well as common group memberships.

We discuss the main informal notions of common ground in the next section. Thereafter, we describe how we had previously built on modal logics to formalize these notions. From there, we demonstrate how to use the described logic to model the representation and usage of common ground in the interaction between human operators and software systems in spacecraft operations. We then discuss how to implement these software systems to consider common ground.

## II. An Informal Model of Common Ground

A basic notion of common ground is proposed by Stalnaker:

**Definition 1** (Common ground as common belief). Stalnaker (Stalnaker, 2002) defines common ground as “*the mutually recognized shared information in a situation, in which an act of trying to communicate takes place*”. Stalnaker proposes one definition of common ground as simply common belief: “*In the simple picture, the common ground is just common or mutual belief*”; and then further proposes that “*a proposition  $\varphi$  is common belief of a group of believers if and only if all in the group believe that  $\varphi$ , all believe that all believe it, all believe that all believe that all believe it, etc.*”

There are multiple issues with this definition. The issue we focus on here is that common belief necessarily implies that everyone in the group actually believes what is common ground. Now take as an example that it is common ground among a government that economic reforms are necessary. But does this imply that every government member actually believes this? This is certainly not the case in our experience. Instead, there is a commonly accepted truth, despite divergent individual beliefs.

An alternative notion of common ground, also put forward by Stalnaker, is based on what the individuals in the group individually accept and commonly belief to be accepted. Acceptance is a mental attitude similar to belief. However, an agent can accept a sentence  $\varphi$  without actually believing it.

**Definition 2** (Acceptance). Hakli (2006) defines acceptance of a proposition  $\varphi$  as: “*a kind of mental act, a decision to treat  $\varphi$  as true in one’s utterances and actions, or an act of adopting a policy to use  $\varphi$  as a premise in one’s theoretical and practical reasoning. This is usually taken to include assuming  $\varphi$  for the sake of some practical purpose, pretending that  $\varphi$  is true or acting as if  $\varphi$  were true, because it is usually allowed that one can simultaneously accept that  $\varphi$  and believe that not  $\varphi$ .*”

Thus, we can accept things without believing them, perhaps for the sake of argument, or to make our current task easier. An example thanks to Stalnaker (2002), is of two acquaintances, Alice and Bob, talking at a party. Alice says to Bob: “*the man drinking the martini is a philosopher*”, referring to a man who Bob knows is actually drinking water from a cocktail glass. Bob accepts that the man is drinking a martini because the contents in the glass are not pertinent to the conversation, and he thinks that correcting Alice

on the matter would disrupt the conversation. Bob accommodates Alice’s belief not by changing his own beliefs on the contents of the glass, but by accepting that the glass contains a martini.

**Definition 3** (Common ground as collective acceptance). [Stalnaker \(2002\)](#) defines a more sophisticated version of common ground as a form of collective acceptance: “*It is common ground that  $\varphi$  in a group if all members accept that  $\varphi$ , and all believe that all accept that  $\varphi$ , and all believe that all believe that all accept that  $\varphi$ , etc.*” This is very similar to Tuomela’s notion of collective acceptance, who defines the base case of collective acceptance of a proposition  $\varphi$  as a situation in which: “*each person comes to accept  $\varphi$ , believes that the others accept  $\varphi$ , and there is a mutual [common] belief about the participants’ acceptance of  $\varphi$* ” ([Tuomela, 2003](#)).

This definition of common ground fits better in our example of economic reforms: a government has a common ground that economic reforms are necessary, and will therefore use this to derive policy, but individuals within the government do not necessarily believe that the reforms are necessary.

Many authors consider that acceptance, and as a result, common ground, are context-dependent; that is, they are only valid within a certain context, for example the current joint activity of a group ([Hakli, 2006](#)). [Kashima et al. \(2007\)](#) note that individuals engaging in a joint activity create common ground relevant to achieving the joint activity, what we shall here call *salient common ground*. Salient common ground is, however, not only created from the information exchanged during the joint activity, what we call *activity-based common ground*, but also from what [Clark \(1996\)](#) calls *personal common ground* and *communal common ground*. Personal common ground comprises the knowledge, belief, and acceptances that are mutually shared by the group due to previous common experiences. Communal common ground comprises the information that is shared because of some common collective identity, such as their nationality, affiliation to a particular organization or political party, occupation, education, recreational activities, or ethnicity, despite not knowing each other and having no reason to believe that someone accepts a proposition except that they identify with the community. Personal and communal common ground themselves are established and maintained mainly through joint activities, in addition to the influence of phenomena such as mass media.

Based on Stalnaker and Tuomela and the observation that acceptance and common ground depend on context, we define common ground as follows ([Pfau et al., 2014](#)):

**Definition 4** (Common ground as context-dependent collective acceptance). *There is a collective acceptance of a proposition  $\varphi$  in a context  $c$  if and only if everyone involved in that context a) accepts  $\varphi$  in the context and b) commonly believes that everyone accepts  $\varphi$  in the context.*

A context can be either a joint activity (for activity-specific common ground) or a group (personal and communal common ground). It is the notions of activity-specific, personal, and communal common ground that demonstrate the importance of acceptance as part of common ground. A definition of common ground as common belief does not permit such notions, because it is not reasonable to assume that one can hold conflicting beliefs at the same time as part of two different contexts. However, it is quite natural to model conflicting *acceptances* as part of different contexts.

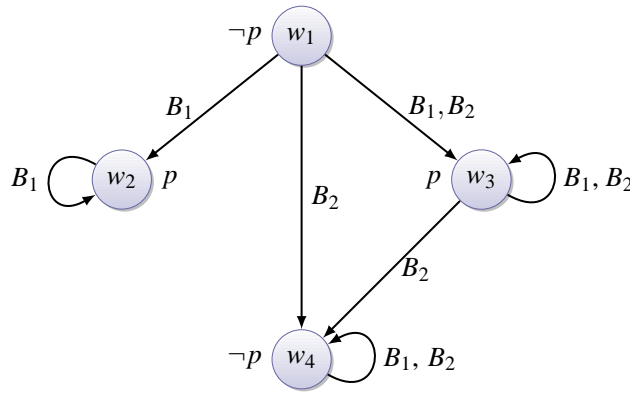
A further observation that we will only touch on in this paper is that the participants in a joint activity draw on some combination of different common grounds: their activity-specific common ground as well as their personal and communal common ground.

### III. A Modal Logic of Common Ground

In this section, we describe a formal model of common ground based on standard modal logics of belief and knowledge, so called *doxastic* or *epistemic* modal logics. This model, which we introduced in recent work ([Pfau et al., 2014](#)), builds on the informal descriptions presented in the previous section. This model of common ground is based on basic modal operators for the mental attitudes (beliefs and acceptances) of individual agents.

A formal model of common ground can be used in at least two ways:

1. As a specification for the representation of common ground in software systems.
2. As a tool for reasoning about software systems that represent common ground.



**Figure 1.** An example of a Kripke model for a doxastic modal logic. On this figure, a label  $B_i$  on an arrow from world  $u$  to  $v$  denotes  $(u, v) \in R_i$ .

This paper addresses the first type of usage. Since the common ground model is based on standard modal logics of belief it lends itself to an integration into systems that are based on these types of specifications, in particular logic-based agent systems (Wooldridge, 2009).

### III.A. Doxastic Modal Logics

The history of modal logics as a tool for describing epistemic and doxastic mental attitudes goes as far back as Hintikka (1962). Since then, the subject has been studied extensively, including the representation of multi-agent group attitudes such as common knowledge and common belief (Fagin et al., 1995).

The syntax of a multi-agent doxastic modal logic consists of: a set  $\Sigma$  of propositional symbols (usually  $p$ ,  $q$ ,  $r$ , etc), which represent basic facts about the environment; the standard propositional connectives ( $\neg$ ,  $\vee$ ,  $\wedge$ , and  $\rightarrow$ ); and a set of *modal operators*  $B_i$  for each agent  $i$ , which describes the beliefs that the particular agent  $i$  has. Compound propositions (usually denoted  $\varphi$  or  $\psi$ ) can be built from this language.

As an example, consider the proposition  $B_a(\text{level}(\text{propellant}, \text{low}) \rightarrow \text{warn}(\text{op}, \text{level}(\text{propellant}, \text{low})))$ , which states that agent  $a$  believes that if the level of the propellant is low, then the operator  $\text{op}$  should be warned of this. Here, the modality  $B_a$  represents agents  $a$ 's beliefs about its world.

Modalities can be nested, thus expressing such statements as  $B_a B_{op} \text{valve\_open} \rightarrow B_a \text{valve\_open}$ , which specifies that if agent  $a$  believes that the operator  $\text{op}$  believes that the valve is open, then agent  $a$  also believes this.

With this language, one can describe the basic axioms of doxastic mental attitudes and, using logical inference, derive the properties that follow from those axioms.

We require a corresponding semantics to give meaning to the sentences of the language. In fact, it is the semantics that enables logical reasoning on the syntactical level at all. The semantics of modal logics are usually based on Kripke semantics (Kripke, 1963). A *Kripke model* consists of a non-empty set  $W$  of *possible worlds* (usually denoted  $u$ ,  $v$ ,  $w$ ), a belief *accessibility relation* for each agent defined over these possible worlds (usually  $R_i \subseteq W \times W$ ), and a *valuation function* that determines which propositional symbols are true in each world (usually  $V : W \rightarrow 2^\Sigma$ ).

Figure 1 shows a clarifying example Kripke model. There are four possible worlds, marked  $w_i$ . The valuation  $V(w_i)$  of each world  $w_i$  denotes whether  $p$  is true in that world or not. For example,  $V(w_1) = \{\}$ ,  $V(w_2) = \{p\}$ . That is,  $p$  is true in  $w_2$  but not in  $w_1$ . The accessibility relations  $R_i$  for each agent (in this case there are two) are denoted by arrows. A label  $B_i$  on an arrow from world  $u$  to  $v$  denotes that  $(u, v) \in R_i$ .

Informally, if we are in a world  $u$ , then the set  $\{v \mid R_i(u, v)\}$  denotes the worlds that agent  $i$  considers *possible* descriptions of the world. In the figure, agent 1 considers the worlds  $w_2$  and  $w_3$  possible in  $w_1$ . Because  $p$  is true in both  $w_2$  and  $w_3$ , agent 1 believes that  $p$  does hold, i.e.  $B_1 p$  in world  $w_1$ . The reason is that the agent does not consider any other world possible that contradicts with this. Note that we cannot say this about agent 2 in world  $w_1$ , because  $w_3$  and  $w_4$  are possible for agent 2 and  $p$  is true in  $w_3$  but not in  $w_4$ . Note that  $B_1 p$  holds also in  $w_2$  and  $w_3$ . Because these are the only two worlds agent 1 considers possible in world  $w_1$ , we also have  $B_1 B_1 p$ , i.e. in world  $w_1$  agent 1 believes that it believes  $p$ .

Depending on additional constraints put on the accessibility relations, particular properties of belief follow. From these constraints, we can prove axioms on the syntactic side, which can be used for further

logical reasoning as indicated above.

When we describe the common ground logic in the next section, we omit the full description of the semantics. However, we believe that it is valuable for the reader to get an idea about the type of semantics we rely on.

### III.B. Syntax

Let  $\Sigma$  be a set of propositional symbols, and  $Ag$  be a finite and non-empty set of agents. We define the common ground modal language  $\mathcal{L}_{CG}$  to be the smallest set that is determined by the following Backus Naur Form:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid B_i\varphi \mid EB_G\varphi \mid CB_G\varphi \mid \mathcal{A}_i^c\varphi \mid EA_c\varphi \mid CA_c\varphi$$

where  $p$  is any propositional symbol in  $\Sigma$ ,  $i$  is any agent in  $Ag$ ,  $G$  is a group of agents such that  $G \subseteq Ag$ , and  $c$  is a *context*. A context is either an activity (denoted by  $\alpha$ ) or a group (denoted by  $G$ ).

Informally,  $B_i\varphi$  specifies that agent  $i$  believes  $\varphi$ ,  $EB_G\varphi$  specifies that  $\varphi$  is believed by every agent in the group  $G$ , and  $CB_G\varphi$  specifies that  $\varphi$  is common belief for the group of agents in  $G$ . Negation is represented using the symbol  $\neg$ , and conjunction with  $\wedge$ . This syntax follows existing modal logics of belief and common belief (Fagin et al., 1995). The proposition  $\mathcal{A}_i^c\varphi$  specifies that agent  $i$  accepts  $\varphi$  in context  $c$ , representing the notions of individual acceptance in joint activity contexts, personal groups, and communal groups. Informally,  $EA_c\varphi$  specifies that every agent involved in context  $c$  accepts  $\varphi$  in that context, and  $CA_c\varphi$  specifies that it is collectively accepted by every agent in context  $c$  that  $\varphi$  holds in that context. Standard propositional connectives such as  $\vee$ ,  $\rightarrow$ , and  $\leftrightarrow$  can be defined in terms of  $\neg$  and  $\wedge$ ; e.g.  $p \vee q$  is the same as  $\neg(\neg p \wedge \neg q)$ .

Recall that context-dependent collective acceptance as denoted by  $CA_c\varphi$  reflects activity-specific, personal, and communal common ground, depending on the type of context  $c$  (joint action or group). The concepts of personal common ground and communal common ground are treated the same, and it is the identification of the group as personal or communal that determines which type of common ground it is.

Consistent with our discussion in the previous section, individual beliefs are not indexed by a context: beliefs are independent of context (although they can result from context-dependent activities).

### III.C. Axiomatics

We describe an axiom schema for the  $\mathcal{L}_{CG}$  logic in this section. While doing so, we omit the detailed semantics and instead build on the intuitive meanings of sentences. The full detail of the logic is reported elsewhere (Pfau et al., 2014). However, we note that the logic's semantics rely on accessibility relations for the belief and acceptance modalities for each agent and context.

The axioms and inference rules listed in the following subsections comprise a Hilbert-style deductive proof system for  $\mathcal{L}_{CG}$ . For the sake of clarity, we describe the axiom schema in two separate sections. The first section describes axioms and inference rules related to belief. The second section describes axioms and inference rules related to acceptance.

#### III.C.1. Individual and Common Belief

We include the following axioms related to belief in the axiom schema:

- (**Prop**) Axioms for propositional logic
- (**K<sub>B</sub>**)  $B_i(\varphi \rightarrow \psi) \rightarrow B_i\varphi \rightarrow B_i\psi$  (Distribution axiom)
- (**D<sub>B</sub>**)  $B_i\varphi \rightarrow \neg B_i\neg\varphi$  (Consistency axiom)
- (**4<sub>B</sub>**)  $B_i\varphi \rightarrow B_iB_i\varphi$  (Positive introspection axiom)
- (**5<sub>B</sub>**)  $\neg B_i\varphi \rightarrow B_i\neg B_i\varphi$  (Negative introspection axiom)
- (**EB**)  $EB_G\varphi \leftrightarrow \bigwedge_{i \in G} B_i\varphi$  (Shared belief axiom)
- (**CB**)  $CB_G\varphi \leftrightarrow EB_G(\varphi \wedge CB_G\varphi)$  (Common belief axiom)

In addition, we account for the following inference rules.

(MP)	$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi}$	(Modus ponens)
(Nec <sub>B</sub> )	$\frac{\varphi}{B_i \varphi}$	(Belief necessitation)
(Ind <sub>CB</sub> )	$\frac{\varphi \rightarrow EB_G(\psi \wedge \varphi)}{\varphi \rightarrow CB_G \psi}$	(Induction of common belief)

The axioms for propositional logic (**Prop**), inference rules *modus ponens* (**MP**) and *necessitation* (**Nec<sub>B</sub>**), and the *distribution* axiom (**K<sub>B</sub>**) define a normal modal logic.

Axiom **K<sub>B</sub>** entails that if an agent believes an implication, then when the agent believes the antecedent of the implication, it also believes the consequent. Axioms **D<sub>B</sub>**, **4<sub>B</sub>**, and **5<sub>B</sub>** further define a standard KD45 logic (Fagin et al., 1995). **D<sub>B</sub>** ensures the *consistency* of beliefs (an agent cannot believe something and its negation), and **4<sub>B</sub>** and **5<sub>B</sub>** represent *positive introspection* and *negative introspection* for beliefs respectively. That is, if an agent believes (does not believe) something, it believes that it believes (does not believe) it.

Axiom **EB** defines how shared belief is derived from individual belief, and axiom **CB** is the so-called *fixed-point axiom* for common belief, which demonstrates the infinite nature of common belief.

The inference rule **MP** defines the rule of modus ponens. Together with the distribution axiom **K<sub>B</sub>**, this implies that an agent believes all the logical consequences of its beliefs. The rule **Nec<sub>B</sub>** specifies that if something is a tautology, the agent believes it. These last two assumptions are known as *logical omniscience* and are commonly accepted to be too strong for resource-bounded agents. The final rule, **Ind<sub>CB</sub>**, describes how common belief is inferred from shared belief.

### III.C.2. Individual and Collective Acceptance

We include the following axioms related to acceptance. In the following, we use the term  $X(c)$  to denote the set of agents that are involved in a context  $c$ :

(K <sub>A</sub> )	$A_i^c(\varphi \rightarrow \psi) \rightarrow A_i^c \varphi \rightarrow A_i^c \psi$	(Distribution axiom)
(D <sub>A</sub> )	$A_i^c \varphi \rightarrow \neg A_i^c \neg \varphi$	(Consistency axiom)
(4 <sub>A</sub> )	$A_i^c \varphi \rightarrow A_i^c A_i^c \varphi$	(Positive introspection axiom)
(5 <sub>A</sub> )	$\neg A_i^c \varphi \rightarrow A_i^c \neg A_i^c \varphi$	(Negative introspection axiom)
(4 <sub>AB</sub> )	$A_i^c \varphi \rightarrow B_i A_i^c \varphi$	(Positive awareness axiom)
(5 <sub>AB</sub> )	$\neg A_i^c \varphi \rightarrow B_i \neg A_i^c \varphi$	(Negative awareness axiom)
(EA)	$EA_c \varphi \leftrightarrow \bigwedge_{i \in X(c)} A_i^c \varphi$	(Shared acceptance axiom)
(CA)	$CA_c \varphi \leftrightarrow EA_c \varphi \wedge CB_{X(c)} EA_c \varphi$	(Collective acceptance axiom)

For the inference rules, we add the following:

(Nec <sub>A</sub> )	$\frac{\varphi}{A_i \varphi}$	(Necessitation)
---------------------	-------------------------------	-----------------

Axioms **K<sub>A</sub>**, **D<sub>A</sub>**, **4<sub>A</sub>**, and **5<sub>A</sub>** together form a standard KD45 modal logic, and therefore the notion of acceptance is equivalent to the notion of belief, but defined over a different accessibility relation.

Axioms **4<sub>AB</sub>** and **5<sub>AB</sub>** specify that an agent has positive and negative awareness about their acceptances. That is, if an agent accepts (does not accept) a proposition, then they believe that they accept (do not accept) that proposition.

Axiom **EA** is the counterpart to axiom **EB** and defines how shared acceptance is derived from individual acceptances. Axiom **CA** defines how collective acceptance is derived from shared acceptance and the common belief of shared acceptance. Note that this formalization of collective acceptance reflects the informal description from Definition 4.

In Section V.B, we present a model of this defined in an agent programming language called *AgentSpeak*. This model uses the axioms above to infer new statements about the beliefs and acceptances of agents and groups of agents.



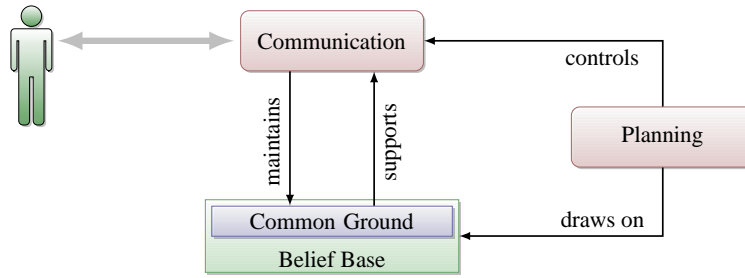


Figure 2. An abstract architecture for communication between operators and a deep-space probe that is able to represent and reason about common ground with its operators.

## IV. Common Ground in Spacecraft Operations

In this section, we sketch models of common ground in two different scenarios: (1) communication between spacecraft operators and deep-space probes or planetary rovers; and (2) simulation of mixed human-agent-robot teams.

### IV.A. Scenario I: Deep-space Robotic Exploration

Among the main challenges for communication between robotic interplanetary probes and mission control centers are:

1. Asynchronous communication models with long transmission times and interrupted connections.
2. The lack of physical co-presence and the resulting difficulty of maintaining shared situational awareness.
3. A large number of scientific and operational stakeholders involved in the operation of the probe.

Based on these challenges, we can define some of the generic requirements of a probe's communication system as follows:

1. Reduce the number and size of data transmissions.
2. Enable the probe to engage actively and purposefully in the maintenance of shared situational awareness.
3. Enable the probe to interact with different stakeholders simultaneously while coping with potentially conflicting stakeholder interests.

We believe that a communication system based on the model of common ground described above achieves these requirements. The key feature of such a communication system, as we will see, is an increased interdependence of the probe and its operators.

Figure 2 shows the abstract architecture of a probe that supports the representation of and reasoning about common ground. At the core of this architecture is a module that allows the probe to represent information about its environment. In artificial intelligence, this module is typically called *belief base*, denoting the set of beliefs a software agent has about the state of the environment. Part of this belief base is the probe's representation of common ground of various contexts. Representing common ground in the same way as the rest of the belief base allows a neat integration. Using the language introduced in the previous section, the representations held in the common ground module of a probe  $P$  are of the type  $B_P C A_c \varphi$ , i.e. they represent the probe's beliefs about the common ground held in different contexts. Logical reasoning over the belief base, in line with the inference rules described in Section III, needs to be supported by suitable inference algorithms.

A communication module takes care of communicating with operators. One of the main responsibilities of this module is to maintain the representation of common ground. Clark (1996) calls the process of establishing and maintaining common ground during a joint activity the *grounding process*, which is necessary for the successful execution of joint activities. At the same time, the communication module draws on the



representation of common ground to determine the information to be transmitted. We will see an example of this below.

The communication module itself is controlled and invoked by the internal planning processes of the probe whenever communication is required. These planning processes also determine the behavior of the probe in the context of the joint activities with the operators. The planning processes themselves draw on the content of the belief base, potentially including the representation of common ground.

Without specifying the details of this architecture, let us provide a few examples how such a system can support achieving the requirements described above.

**REDUCE THE NUMBER AND SIZE OF DATA TRANSMISSIONS** Based on the assumption that what is represented as common ground is information that is actually shared, the communication system can make informed decisions about which information can be omitted from transmission. This can lead to a reduction in data transmission. Assume as a simple example that the probe  $P$  has encoded the following about its common ground with a particular operator  $O$  that is part of its team:  $CA_{\{P,O\}}\varphi \rightarrow \psi$ . Now, if the probe learns that  $\varphi$ , it is sufficient from the probe's perspective to establish  $CA_{\{P,O\}}\varphi$  since  $CA_{\{P,O\}}\psi$  follows through modus ponens. The probe no longer needs to query the operator to ask if  $\psi$  holds.

**MAINTAIN SHARED SITUATIONAL AWARENESS** Common ground plays a key role in the shared situational awareness (Saner et al., 2009) between the probe and its operators. By treating common ground as a first-class citizen of the probe's representation of the environment, maintaining shared situational awareness becomes an inherent feature of the probe's activities. Moreover, maintaining common ground is a direct effect of communication, and communication is a natural by-product of the probe's planning activities. If common ground is represented symbolically as we assume here, the probe can improve shared situational awareness by presenting its assumptions about common ground to the operators, based on the symbolic form it is represented in internally. A key issue in human-robot interaction is the difficulty for operators to understand the robots' reasoning (Stubbs et al., 2007), which can be avoided by making the robot more transparent. To detect and mitigate mis-communication, the probe can represent the operators' perception of common ground on top of its own perception (that is, what it believes is the operator's belief of common ground), using the same representational system. This can help the probe to disentangle the discrepancies between actual common ground and perceived common ground.

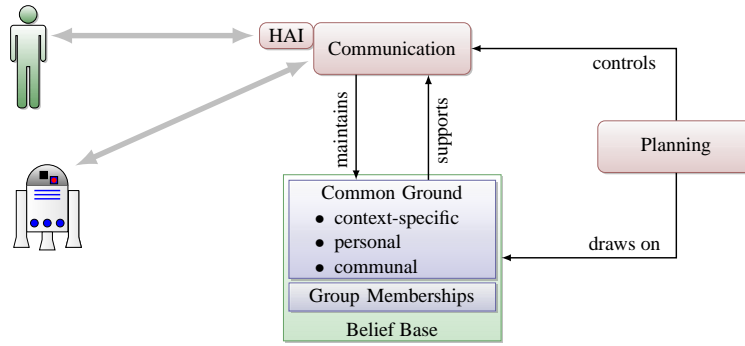
**INTERACT WITH DIFFERENT STAKEHOLDERS SIMULTANEOUSLY** The model of common ground introduced in the previous section can represent different and possibly conflicting common ground associated with different contexts, i.e.  $CA_{c_1}\varphi \wedge CA_{c_2}\neg\varphi$ . The key to this capability, which is not found in other models of common ground, is the reliance on the notion of acceptance. While interacting with different parties, the probe can accept conflicting information until eventually there is a necessity to resolve the conflict. For example, the probe could establish common ground with scientists to point an instrument at a particular attitude while also establishing with engineers to go into maintenance mode at the same time. Obviously, this conflict requires resolution through communication but this could deliberately be deferred to a later point.

Overall, representing and using common ground improves the interdependence between probes and their operators: probes adjust their communication to the current situation and the information assumed to be shared to reduce data transmission, while still maintaining sufficient shared situational awareness.

Stubbs et al. (2008) describe the design of an Earth-based proxy that, on behalf of the deep-space probe, establishes common ground about planned activities with the operators. Experiments showed that, because the interaction with the proxy was more immediate, operators developed a better sense of interdependence with the system and also developed more effective plans. Stubbs et al. build on an ad-hoc model of common ground while here we demonstrate the use of a carefully crafted model. The same restriction applies to work by Heikkilä (2010) on enabling robots to represent their common ground with operators.

#### IV.B. Scenario II: Simulation of Mixed Human-Agent-Robot Teams

Smets et al. (2010) conduct simulations of the manned spaceflight working environment and its actors to evaluate a virtual crew assistant. They use simulations instead of experiments for their evaluation because astronauts are hard to come by and the real environment might not be accessible for experiments (e.g. the



**Figure 3. An abstract architecture for a virtual team member in simulations of a mixed human-agent-robot environment.**

Martian surface). The simulation model is based on the multi-agent Brahms language for the modeling and simulation of human work practices (Sierhuis et al., 2009).

Amongst others, we see two key requirements that such a simulation system needs:

1. Agents that represent virtual team members must communicate seamlessly with other agents as well as with potential humans stepping in for virtual team members.
2. Agents need to be modeled with sufficient fidelity, in particular their communication with human team members.

We believe that adhering to the common ground model described above and to a suitable grounding mechanism are crucial to achieving these requirements.

We extend the detail of the agent architecture described in the previous section (in Figure 2) and discuss how these details help to achieve the requirements listed above. The refined architecture is shown in Figure 3. First, we add a communication interface that renders human communication compatible with inter-agent communication, from the perspective of an agent (called *HAI* for “human-agent interface” in the figure). Second, we refine the belief base with information about the different types of common ground and group memberships. These changes support the identified requirements as follows.

**COMMUNICATE SEAMLESSLY WITH OTHER AGENTS AND HUMAN TEAM MEMBERS** We assume that an agent can communicate with other agents natively, for example via agent communication languages such as KQML (Finin et al., 1994) or FIPA (O’Brien and Nicol, 1998). Communication with human team members is likely more complex and requires a different interface. Here we assume that this human-agent interface is attached directly to the agent’s communication module. This allows for tighter control from the agent’s side, but also requires the agent to know whether a communication partner is virtual or human, in order to decide whether the human-agent interface needs to be deployed. As an alternative, the human-agent interface could reside on the human’s side such that any communication from the human with the agent system is directly translated into an agent compatible format. However, reducing communication to an agent communication language might also imply a cut in information. Either way, introducing the human-agent interface enables agents to interact with human team members, in addition to virtual team members.

**MODEL COMMUNICATION WITH SUFFICIENT FIDELITY** Common ground and the grounding process play a key role in human-human communication. Therefore, agents in this system also need a sufficiently accurate model of grounding and common ground for their interactions with human team members. The model of common ground we describe in Section III has been carefully created based on philosophical and psychological precursor literature. We briefly describe how some of the detail in the model contributes to an accurate representation of the mixed human-robot-agent team environment.

First, we discussed earlier that common ground comes in different types; and the common ground assumed in a particular joint activity consists not only of the common ground created immediately in that activity (activity-specific common ground) but also of the common ground that is held because of the previous common experiences (personal common ground) and common group memberships of the participants (communal common ground). Concretely in this scenario this means, for example, that a spacecraft operations engineer (SOE) agent would draw on the communal common ground of SOEs in interactions with other

SOEs. Likewise, the agent would draw on the personal common ground created during previous interactions with its communication partners. A requirement for the representation of communal common ground is a representation of group memberships to be included in the belief base. Also, a suitable model needs to be created for the “merging” of activity-specific, personal, and communal common ground.

Second, one purpose of the grounding process is to align perceived and actual common ground. For agents to track discrepancies between their actual common ground and the common ground that they each assume to hold they need to be able to represent what they believe about the perception of common ground of other agents. We mentioned this in the context of the previous scenario. Only a model of common ground as differentiated as the one we build on here can support this requirement.

Overall, the detail in the common ground model supports the simulation model to represent the interdependence inherent to human joint activities more accurately.

## V. Representing Common Ground in Software

In this section, we explore how a representation of common ground can be implemented in agent systems such as the ones envisioned in the previous section.

First, we need to be more precise about an agent’s belief base, which we only touched upon in the previous section. The reason is that the common ground an agent ascribes to a particular context is part of the environment and hence to be represented in the agent’s belief base.

A belief base is described by a formal knowledge representation language  $\Sigma$ , an inference relation  $\models$ , and a belief update rule (Dastani et al., 2009). The content of the belief base is a set of sentences in the language  $\Sigma$ . The inference relation essentially applies deduction rules to determine whether a given query is entailed by the belief base. Normally, the deduction rules are required to be sound with respect to the semantics of the language  $\Sigma$  and to be efficiently computable (Konolige, 1983). The belief update rule determines how new beliefs are to be merged into the belief base.

The inference relation may be incomplete, either because the deduction rules are too weak or because of limitations of the inference procedure itself (Konolige, 1983). Incompleteness reflects the fact that an agent is resource-bounded and cannot reasonably be required to derive all logical consequences of its beliefs. To avoid complications it is advisable to disregard limitations of the inference procedure and require the beliefs that can be derived to be closed under deduction as performed by the inference procedure (although the deduction rules might be incomplete).

While the logic that we have defined for common ground (Pfau et al., 2014) is both sound and complete, this is not sufficient for implementing real systems. First, the major problem is that checking entailment of a formula in this logic is PSPACE-complete (Halpern and Moses, 1992). Second, the logic requires the developer to specify accessibility relations for all worlds, which is quite unnatural compared to keeping track of beliefs using symbolic formula.

Therefore, we seek to represent a tractable subset of the common ground logic in an of-the-shelf symbolic belief base implementation. We opt for the agent programming language AgentSpeak, which, amongst others, defines an agent’s belief base (Rao, 1996). The reasons for this decision are that:

- AgentSpeak is a popular agent programming language.
- AgentSpeak implements the common belief-desire-intention agent model, which has been researched extensively (Bratman, 1987).
- Jason, as a full-fledged AgentSpeak interpreter, is freely available (Bordini et al., 2007).

We briefly describe relevant parts of AgentSpeak and Jason and then present how the belief base of a Jason agent can represent selected parts of the common ground logic described earlier.

### V.A. AgentSpeak and Jason

In the following, we rely on Jason’s extended version of AgentSpeak. The main part of the specification of a Jason agent is the specification of its belief base. The parts of the language,  $\Sigma$ , for describing a Jason agent’s belief base that are relevant to this paper are illustrated by the following example (Bordini et al., 2007):

```

1 // facts
2 solar_panels([panel1, panel2, panel3]).
3 deployed(panel1).
4
5 // rules
6 solar_panels_deployed :- solar_panels(Panels) & solar_panels_deployed(Panels).
7 solar_panels_deployed([Head|Tail]) :- deployed(Head)
8                                     & .length(Tail, Length)
9                                     & Length == 0
10 solar_panels_deployed([Head|Tail]) :- deployed(Head)
11                                     & .length(Tail, Length)
12                                     & Length > 0
13                                     & solar_panels_deployed(Tail).

```

An agent's belief base consists of facts and rules. Facts are basic beliefs that the agent believes true of the world. Each fact is a literal consisting of an atom followed by a list of terms in brackets. In the example above, the agent believes that there is a set of three solar panels and that the first panel is deployed. Note the support of lists. New beliefs can be added by the agent at runtime, triggering the belief update function.

Rules determine the deductions that the inference relation can apply. The right-hand side of the rule is the antecedent, the left-hand side is the consequent. If the antecedent is provable from the agent's belief base, so is the consequent. Note that a rule essentially reflects a logical implication and an application of modus ponens to this implication. Variables are allowed in rules (distinguished by a capital initial letter) and unification is applied during deduction to resolve these variables. The rules in the example above allow the agent to answer a query against `solar_panels_deployed` to determine whether all solar panels have been deployed yet.

In the same way as Prolog, Jason normally takes the closed-world assumption and assumes negation of a fact to be given when the inference rule fails to deduce that fact from the belief base. Anything that is not explicitly represented in the belief base is assumed to be false. This is called *negation as failure*. Alternatively, the open-world assumption can be taken by using the symbol “ $\sim$ ” to flag facts that are explicitly believed to be false by the agent. In this case, failing to derive a sentence  $\varphi$  and its negation  $\sim\varphi$  means that the agent is ignorant about  $\varphi$ .

## V.B. Common Ground in AgentSpeak

We seek to model the axioms of the common ground logic described in Section III in AgentSpeak to render Jason agents capable of reasoning about common ground. AgentSpeak has a few limitations that make this task non-trivial:

1. Nested modalities are not supported natively. In fact, only one level of belief modality is supported, namely that one that is implicitly attached to each item in the agent's knowledge base. However, literals and terms can be nested arbitrarily. For example, the following is a valid fact in a Jason belief base:

```
bel(agent1, bel(agent2, ~bel(agent1, p))).
```

and represents that **agent1** believes that **agent2** believes that **agent3** does not believe  $p$ .

We will make use of this to represent nested belief and acceptance modalities.

2. There is no syntactical equivalent to implication. Rules correspond implicitly to applications of modus ponens to implications. However, there is no way to represent implication nested within literals. Note that this would be necessary to represent the distribution axioms  $\mathbf{K_B}$  and  $\mathbf{K_A}$ . In fact, there is no possibility to represent any logical expressions nested within literals.
3. Both **CB** and **CA** imply the *infinite* nesting of modalities, which AgentSpeak does not foresee and the Jason interpreter cannot deal with.

**BELIEF** Table 1 illustrates the translation of axioms related to belief into AgentSpeak. The main prerequisite is that we introduce a literal `bel(Ag, P)` that represents the modality of agent **Ag** believing the sentence  $P$ . This enables us to represent the following nested modalities:

```
belief ::= [~]bel(agent, belief) | literal
```

Property	Axiom	AgentSpeak
<b>Prop</b>	Axioms of prop. logic	—
<b>K<sub>B</sub></b>	$B_i(\varphi \rightarrow \psi) \rightarrow B_i\varphi \rightarrow B_i\psi$	—
<b>D<sub>B</sub></b>	$B_i\varphi \rightarrow \neg B_i\neg\varphi$	$\sim\text{bel}(\text{Ag}, \sim P) :- \text{bel}(\text{Ag}, P)$
<b>4<sub>B</sub></b>	$B_i\varphi \rightarrow B_i B_i\varphi$	$\text{bel}(\text{Ag}, \text{bel}(\text{Ag}, P)) :- \text{bel}(\text{Ag}, P)$
<b>5<sub>B</sub></b>	$\neg B_i\varphi \rightarrow B_i\neg B_i\varphi$	$\text{bel}(\text{Ag}, \sim\text{bel}(\text{Ag}, P)) :- \sim\text{bel}(\text{Ag}, P)$
<b>EB</b>	$EB_G\varphi \leftrightarrow \bigwedge_{i \in G} B_i\varphi$	$\text{bel}(\text{Ag}, P) :- \text{eb}(\text{G}, P) \ \& \ .\text{member}(\text{Ag}, \text{G}).$ $\text{eb}([\text{H} \text{T}], P) :- .\text{length}(\text{T}, \text{L}) \ \& \ \text{L} == 0$ $\quad \quad \quad \& \ \text{bel}(\text{H}, P).$ $\text{eb}([\text{H} \text{T}], P) :- .\text{length}(\text{T}, \text{L}) \ \& \ \text{L} > 0$ $\quad \quad \quad \& \ \text{bel}(\text{H}, P) \ \& \ \text{eb}(\text{T}, P).$
<b>CB</b>	$CB_G\varphi \leftrightarrow EB_G(\varphi \wedge CB_G\varphi)$	$\text{eb}(\text{G}, P) :- \text{cb}(\text{G}, P).$ $\text{bel}(\text{Ag1}, \text{bel}(\text{Ag2}, P)) :-$ $\quad \quad \quad \text{cb}(\text{G}, P)$ $\quad \quad \quad \& \ \text{members}([\text{Ag1}, \text{Ag2}], \text{G}).$ $\text{bel}(\text{Ag1}, \text{bel}(\text{Ag2}, \text{bel}(\text{Ag3}, P))) :-$ $\quad \quad \quad \text{cb}(\text{G}, P)$ $\quad \quad \quad \& \ \text{members}([\text{Ag1}, \text{Ag2}, \text{Ag3}], \text{G}).$ $\dots$ $\text{eb}(\text{G}, \text{cb}(\text{G}, P)) :- \text{cb}(\text{G}, P).$

Table 1. AgentSpeak representation of axioms related to belief.

Propositional axioms that can be represented by AgentSpeak are implemented by the Jason interpreter. Distribution **K<sub>B</sub>** cannot be represented because of the lack of a syntax for implication. Consistency is implemented directly as follows: If agent **Ag** believes sentence **P**, then the agent does not believe the negation of **P**. The implementation of positive and negative introspection follows the same schema.

For the implementation of **EB** two directions need to be represented: the expansion of **EB** into individual beliefs and the deduction of **EB** from individual beliefs. These correspond to the two directions of the  $\leftrightarrow$  operator. In the latter case, we use recursion to determine whether all members of a given group/list of agents **[H|T]** (**H** unifies with the head of the given list, **T** with its tail) believe that **P**.

In the case of **CB**, we only represent the direction from left to right, i.e. the expansion of common belief to individual beliefs. This is because the right-to-left case—deriving common belief from individual beliefs—requires an infinite number of inferences, due to the infinitely long change of nested beliefs.

For the left-to-right case, we represent a bounded level of belief nestings, beginning with the base case  $\text{eb}(\text{G}, P)$ . Further levels need to be represented with individual beliefs. The reason is that the proposition  $\text{bel}(\text{Ag1}, \text{bel}(\text{Ag2}, \text{bel}(\text{Ag3}, \dots)))$  does not automatically follow from  $\text{eb}(\text{G}, \text{eb}(\text{G}, \text{eb}(\text{G}, \dots)))$  because **K<sub>B</sub>** and **Nec<sub>B</sub>** cannot be represented. Jason supports customized pre-processing directives, which allow the expansion of **CB** to be automated. We represent, however, the agents' belief in common belief ( $\text{eb}(\text{G}, \text{cb}(\text{G}, P))$ ).

We could also place a bound on the right-to-left case, such that if a group of agents believe all other agents believe that etc., up to a fixed bound, then we infer common belief. However, we have not represented this here.

We refrain from implementing the inference rules. Modus ponens **MP** can be considered to be supported natively by the Jason interpreter, as discussed above. This implies that any required implication needs to be implemented by suitable rules manually. Belief necessitation **Nec<sub>B</sub>** is not directly supported and needs to be implemented manually from case to case. However, the lack of a syntax for implication renders belief necessitation much less important than it is in common ground logic. There is no way for a Jason agent to believe axioms or theorems that include implication, simply because they cannot be represented. However, the need for manual specifications of modus ponens and necessitation reflect that agents are resource-bounded

Property	Axiom	AgentSpeak
<b>EA</b>	$EA_c\varphi \leftrightarrow \bigwedge_{i \in X(c)} A_i^c\varphi$	$\text{acc}(\text{Ag}, C, P) \text{ :- } \text{.list}(C)$ $\quad \& \text{.member}(\text{Ag}, C)$ $\quad \& \text{ea}(C, P).$
		$\text{acc}(\text{Ag}, C, P) \text{ :- not } \text{.list}(C)$ $\quad \& \text{participants}(C, G)$ $\quad \& \text{.member}(\text{Ag}, G)$ $\quad \& \text{ea}(C, P).$
		$\text{ea}(C, P) \text{ :- } \text{.list}(C)$ $\quad \& \text{ea}(C, C, P).$
		$\text{ea}(C, P) \text{ :- not } \text{.list}(C)$ $\quad \& \text{participants}(C, G)$ $\quad \& \text{ea}(G, C, P).$
		$\text{ea}([H T], C, P) \text{ :- } \text{.length}(T, L)$ $\quad \& L == 0$ $\quad \& \text{acc}(H, C, P).$
		$\text{ea}([H T], C, P) \text{ :- } \text{.length}(T, L)$ $\quad \& L > 0$ $\quad \& \text{acc}(H, C, P)$ $\quad \& \text{ea}(T, C, P).$
<b>CA</b>	$CA_c\varphi \leftrightarrow CB_{X(c)}EA_c\varphi \wedge EA_c\varphi$	$\text{ea}(C, P) \text{ :- } \text{ca}(C, P).$
		$\text{bel}(\text{Ag1}, \text{acc}(\text{Ag2}, C, P)) \text{ :-}$ $\quad \text{.list}(C)$ $\quad \& \text{ca}(C, P)$ $\quad \& \text{members}([\text{Ag1}, \text{Ag2}], C).$
		$\text{bel}(\text{Ag1}, \text{acc}(\text{Ag2}, C, P)) \text{ :-}$ $\quad \text{not } \text{.list}(C)$ $\quad \& \text{ca}(C, P)$ $\quad \& \text{participants}(C, G)$ $\quad \& \text{members}([\text{Ag1}, \text{Ag2}], G).$
		...
		$\text{cb}(C, \text{ea}(C, P)) \text{ :- } \text{ca}(C, P).$
		$\text{ca}(C, P) \text{ :- } \text{cb}(C, \text{ea}(C, P)) \& \text{ea}(C, P).$

Table 2. AgentSpeak representation of axioms related to acceptance.

and can neither believe all the logical consequences of their beliefs nor all theorems. The agent designer can flexibly decide which deduction rules the resource-bounded agent is able to apply (see also our discussion in the beginning of this section). The induction of common belief **Ind<sub>CB</sub>** is omitted because we only represent the expansion of common belief into individual beliefs, not vice versa, and because **Ind<sub>CB</sub>** also requires a representation of implication.

**ACCEPTANCE** The representation of axioms related to acceptance are similar to those of belief. In Table 2 we show the AgentSpeak versions of **EA** and **CA**. Note that the operator for acceptance has an additional argument, namely the context. The representation of **EA** is similar to the one of **EB** with the main difference being the need for representing **EA** for two different types of contexts: a joint activity context with an associated set of participants (**not .list(C)**), and a group context in which the participants are just the members of the group (**.list(C)**). In the case of **CA** we represent both the left-to-right and the right-to-left case. However, the right-to-left case is only supported for the case in which **cb(C, ea(C, P))** and **ea(C, P)** are part of the belief base as syntactic operators (i.e. not in the form of individual beliefs).



DISCUSSION Despite the limitations of the AgentSpeak language, we are able to represent a significant part of the common ground logic in AgentSpeak. The main issues are the lack of a native representation of implication and of nested modalities. We intend to investigate further how these aspects might be represented in AgentSpeak nevertheless. The advantage of Jason is its extensive customizability. As a first approach at the native support of nested modalities we intend to follow Konolige’s (1983) idea to represent the content of each belief modality as a separate belief base. The interpretation of a belief modality at one level then requires the interpretation of the content of this belief modality within a subordinate belief base. Our analysis is relevant to a broader class of agent programming languages as these suffer from a similar set of limitations.

## VI. Conclusions

A key characteristic of space operations is the control of complex technical equipment in highly dynamic and unknown environments. Yet with increasing system complexity, the burden on spacecraft operators will only rise. This is especially the case if these systems are equipped with more autonomy, while not considering how this autonomy affects the human operators. The way forward is to design systems that take a more active role in the support of the operators, including systems that are designed for engaging in joint activities with their users. This would allow these systems to truly collaborate with the operators and it would enable more natural and effective interactions between users and systems.

Many psychologists and philosophers before us have identified the establishing and maintenance of common ground as a crucial aspect in human joint activities, and recent work in human-agent-robot interaction has emphasized the role of common ground. We have recently proposed the first formal models of common ground, which we based on a detailed analysis of philosophical and psychological literature. In this paper, we have projected one of these models onto potential applications in spacecraft operations, and have explored how the model can be implemented computationally and we have discussed potential pitfalls.

The topic of this paper is important for future space software systems in general and systems supporting human-agent-robot teamwork in particular. Our work provides a precursor for the type of systems and domains that might benefit from a representation of common ground, and it also provides a starting point for concrete implementations.

## References

- Allan, K. (2013). What is Common Ground? In Capone, A., Lo Piparo, F., and Carapezza, M., editors, *Perspectives on Linguistic Pragmatics*, chapter 11, pages 285–310. Springer.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons.
- Bradshaw, J. M., Dignum, V., Jonker, C. M., and Sierhuis, M. (2012). Human-agent-robot teamwork. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI ’12*, pages 487–488, New York, NY, USA. ACM.
- Bradshaw, J. M., Feltovich, P., and Johnson, M. (2011). Human-agent interaction. In Boy, G., editor, *Handbook of Human-Machine Interaction*, page 283302. Ashgate.
- Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. Cambridge University Press, Cambridge, UK.
- Clark, H. (1996). *Using language*. Cambridge University Press.
- Dastani, M. M., Hindriks, K. V., Novk, P., and Tinnemeier, N. A. M. (2009). Combining multiple knowledge representation technologies into agent programming languages. In Baldoni, M., Son, T., Riemdsdijk, M. B., and Winikoff, M., editors, *Declarative Agent Languages and Technologies VI*, volume 5397 of *Lecture Notes in Computer Science*, pages 60–74. Springer Berlin Heidelberg.
- Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (1995). *Reasoning about knowledge*, volume 4. MIT press Cambridge.



- Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an agent communication language. In *Proceedings of the Third International Conference on Information and Knowledge Management, CIKM '94*, pages 456–463, New York, NY, USA. ACM.
- Fitts, P. (1951). *Human Engineering for an Effective Air Navigation and Traffic Control System*. National Research Council, Washington, D.C.
- Hakli, R. (2006). Group beliefs and the distinction between belief and acceptance. *Cognitive Systems Research*, 7(2):286–297.
- Halpern, J. Y. and Moses, Y. (1992). A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial intelligence*, 54(3):319–379.
- Heikkilä, S. S. (2010). The role of natural interaction in astronaut-robot cooperation. In *Proceedings of the International Astronautical Congress (IAC)*, Prague, Czech Republic.
- Hintikka, J. (1962). *Knowledge and belief*, volume 414. Cornell University Press, Ithaca.
- Kashima, Y., Klein, O., and Clark, A. (2007). Grounding: Sharing information in social interaction. *Social communication*, pages 27–77.
- Konolige, K. (1983). A deductive model of belief. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'83*, pages 377–381, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Kripke, S. (1963). Semantical considerations on modal logic. *Acta philosophica fennica*, 16(1963):83–94.
- Lee, B. P. (2001). Mutual knowledge, background knowledge and shared beliefs: Their roles in establishing common ground. *Journal of Pragmatics*, 33(1):21–44.
- O'Brien, P. D. and Nicol, R. C. (1998). FIPA - Towards a standard for software agents. *BT Technology Journal*, 16(3):51–59.
- Pfau, J., Miller, T., Sonenberg, L., and Kashima, Y. (2014). Logics for common ground. (Under review).
- Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In Velde, W. and Perram, J. W., editors, *Agents Breaking Away*, volume 1038 of *Lecture Notes in Computer Science*, pages 42–55. Springer Berlin Heidelberg.
- Saner, L. D., Bolstad, C. A., Gonzalez, C., and Cuevas, H. M. (2009). Measuring and predicting shared situation awareness in teams. *Journal of Cognitive Engineering and Decision Making*, 3(3):280–308.
- Sierhuis, M., Clancey, W. J., and Hoof, R. J. J. (2009). Brahms an agent-oriented language for work practice simulation and multi-agent systems development. In El Fallah Seghrouchni, A., Dix, J., Dastani, M., and Bordini, R. H., editors, *Multi-Agent Programming*, chapter 3, pages 73–117. Springer US.
- Smets, N., Bradshaw, J., van Diggelen, J., Jonker, C., Neerincx, M., de Rijk, L., Senster, P., Sierhuis, M., and ten Thijs, O. (2010). Assessing human-agent teams for future space missions. *Intelligent Systems, IEEE*, 25(5):46–53.
- Stalnaker, R. (2002). Common ground. *Linguistics and Philosophy*, 25(5):701–721.
- Stubbs, K., Hinds, P., and Wettergreen, D. (2007). Autonomy and common ground in human-robot interaction: A field study. *Intelligent Systems, IEEE*, 22(2):42–50.
- Stubbs, K., Wettergreen, D., and Nourbakhsh, I. (2008). Using a robot proxy to create common ground in exploration tasks. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 375–382.
- Tuomela, R. (2003). Collective acceptance, social institutions, and social reality. *American Journal of Economics and Sociology*, 62(1):123–165.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons, Hoboken, NJ, 2nd edition.